

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Jakub Hladík



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY

A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

NÁSTROJ PRO NÁVRH HLUBOKÉ NEURONOVÉ SÍTĚ

TOOL FOR DEEP NEURAL NETWORK DESIGN

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jakub Hladík

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Václav Uher

BRNO 2016



Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

Student: Jakub Hladík

ID: 164281

Ročník: 3

Akademický rok: 2015/16

NÁZEV TÉMATU:

Nástroj pro návrh hluboké neuronové sítě

POKyny PRO VYPRACOVÁNÍ:

Cílem práce je vytvořit GUI pro vizualizaci neuronové sítě pro nástroj Caffe vyvíjený skupinou Berkeley Vision and Learning Center. Nástroj bude zobrazovat jednotlivé vrstvy, jejich konfiguraci a vzájemné propojení.

DOPORUČENÁ LITERATURA:

[1] JIA, Yangqing, et al. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the ACM International Conference on Multimedia. ACM, 2014. p. 675-678.

[2] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. 2012. p. 1097-1105.

[3] Java Documentation [online]. 2015. [cit. 2015-10-19]. Dostupné z: <http://docs.oracle.com/en/java/>

Termín zadání: 1.2.2016

Termín odevzdání: 1.6.2016

Vedoucí práce: Ing. Václav Uher

Konzultant bakalářské práce:

doc. Ing. Jiří Mišurec, CSc., předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Cieľom tejto bakalárskej práce bolo vytvoriť program pre vizualizáciu umelých neurónových sietí. V prvej časti sú definované umelé neurónové siete, porovnané s biologickými sieťami a s procesormi osobných počítačov. V stručnosti je popísaná aj ich funkcia a výhody, respektíve nevýhody umelých neurónových sietí. V druhej časti je stručne vysvetlený problém učenia umelých neurónových sietí. Sú tu vysvetlené postupy učenia ako aj modely, ktoré sa pri učení používajú. V tretej časti je spravený úvod do pracovného prostredia Caffe a prehľad najčastejšie vyskytujúcich sa vrstiev, ktoré neurónové siete vytvorené pomocou Caffe obsahujú. V ďalšej časti sa pristupuje k samotnej tvorbe programu. Stručnému rozboru problému ako aj podrobnému opisu postupu pri tvorbe jednotlivých tried programu.

KLÚČOVÉ SLOVÁ

Umelá inteligencia, umelé neurónové siete, systémy hlbokého učenia, grafické rozhranie.

ABSTRACT

The aim of this thesis was to create a program for visualization of artificial neural networks. The first part contains definition of artificial neural networks, comparison with biological neural networks and comparison with processors used in personal computers. Also contains brief description of their function and advantages/disadvantages of artificial neural networks. The second part contains brief explanation of deep learning. There are described methods and models used for learning. The third part contains introduction to Caffe framework and summary of the most frequently occurring layers in neural networks, that are created by using Caffe. Next part contains brief analysis of created program as well as detailed description of classes, that program contains.

KEYWORDS

Artificial intelligence, artificial neural networks, deeplearning systems, GUI.

HLADÍK, Jakub *Nástroj pro návrh hluboké neuronové sítě*: bakalárska práca. Místo: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2016. 38 s. Vedúci práce bol Ing. Václav Uher

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Nástroj pro návrh hluboké neuro-
nové sítě“ vypracoval(a) samostatne pod vedením vedúceho bakalárskej práce, využitím
odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a
uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvo-
rením tejto bakalárskej práce som neporušil(a) autorské práva tretích osôb, najmä som
nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo
majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledu-
júcich autorského zákona č. 121/2000 Sb., o právu autorskom, o právach súvisejúcich
s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskor-
ších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia
časti druhej, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

Miesto

.....

podpis autora(-ky)

POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej práce pánovi Ing. Václavovi Uhrovi, za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

Místo

.....

podpis autora(-ky)

POĎAKOVANIE

Výzkum popsaný v tejto bakalárskej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Místo

.....
podpis autora(-ky)

OBSAH

Úvod	9
1 Teoretická časť študentskej práce	10
1.1 Neurónová sieť	10
1.1.1 Inšpirácia	10
1.1.2 Ako neurónové siete fungujú	11
1.1.3 Využitie neurónových sietí	12
1.1.4 Výhody a nevýhody umelých neurónových sietí	15
1.2 Učenie neurónových sietí	16
1.2.1 Postup učenia	16
1.2.2 Modely učenia	19
1.3 Caffe	24
1.3.1 Úvod do Caffe	24
1.3.2 Prehľad vrstiev	25
2 Výsledky studentské práce	27
2.1 Úvod do problému	27
2.2 Tvorba programu	27
2.2.1 Loader.java	29
2.2.2 LayerVis.java	30
2.2.3 Cir.java	30
2.2.4 GUI.java	30
3 Záver	33
Literatúra	34
Zoznam symbolov, veličín a skratiek	36
Zoznam príloh	37
A Obsah priloženého CD	38

ZOZNAM OBRÁZKOV

1.1	Neurón	10
1.2	Umelý neurón	11
1.3	Smer toku dát a chýb v RNN	23
2.1	Ukážka súboru prototxt	28
2.2	a) riešené tmavým a svetlým kruhom , b) DropShadow	31
2.3	Ukážka programu	32

ÚVOD

Ludský mozog sa skladá z miliárd neurónov, ktoré sú pospájané medzi sebou nervovými vláknami, synapsiami. Spolu tvoria obrovskú sieť, neurónovú sieť, vďaka ktorej sme schopný prežívať emócie, hýbať sa, rozmýšľať a mnohé iné. Hoci už vedci skúmali niektoré aspekty týchto sietí, stále nevieme, ako presne fungujú. Pokiaľ nevieme, ako presne fungujú, nie sme schopný vytvoriť ich biologickú podobu niekde v laboratóriu. No na základe znalostí, ktoré už máme, sme schopný vytvoriť aspoň analógovú podobu neurónových sietí. Takéto, siete voláme umelo vytvorené siete. Čas a investície strávené výskumom umelých neurónových sietí sa vyplatili, čo je možno vidieť už dnes a to v ich aplikácii. Využívajú sa pri automatizácii rozpoznávania obrázkov, výroby, zefektívňovaní výrobných procesov, v zdravotníctve, pri predikcii cien na svetovom trhu, pri rozpoznávaní zvukových signálov, písaného textu, pri tvorbe parkovacích systémov, samočinných automobilov a dokonca aj pri syntetizácii reči na základe predloženého textu.

Sieť sa zväčša skladá z jedného výstupu a viacerých vstupov. Je schopná prevádzať signály zo vstupu, informácie z vonkajšieho sveta, na výstupné, riadiace signály.

Pre správnu funkciu je potrebné obrovské množstvo vstupných dát, ktoré postupne prechádza sieťou, vrstvou po vrstve. Pod pojmom vrstva si môžeme predstaviť zhuk obrovského množstva neurónov, so špecifickou úlohou.

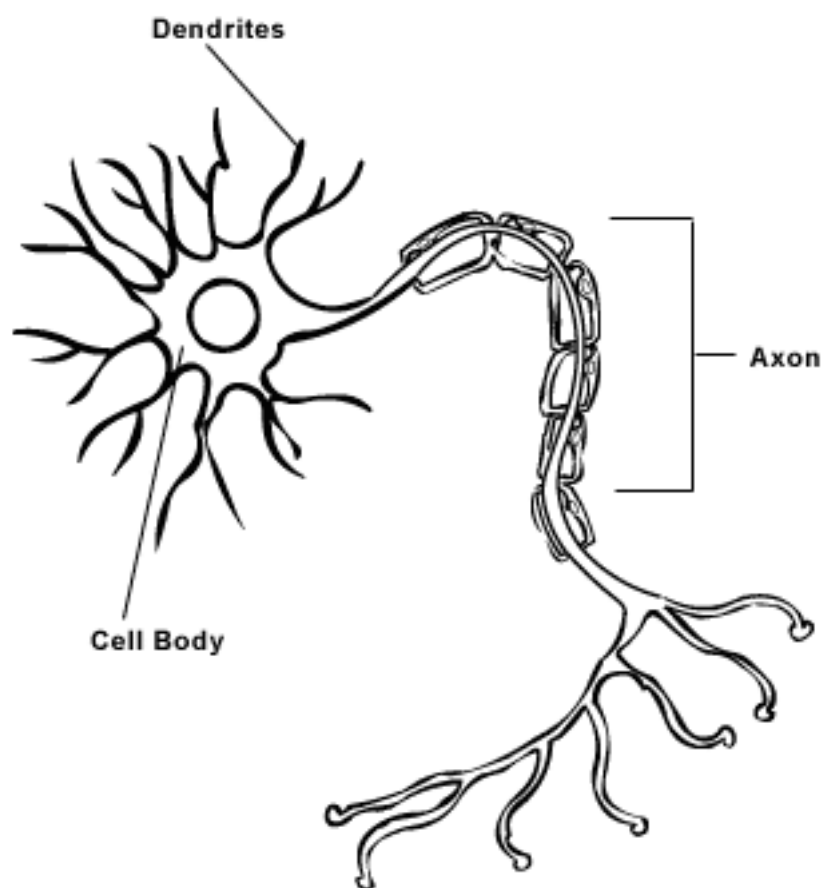
Počet vrstiev ako aj počet neurónov v jednej vrstve je limitovaný len prostriedkami počítača alebo čipu, na ktorých sú umelé neurónové siete realizované. Samozrejme platí pravidlo, čím je sieť zložitejšia, rozsiahlejšia, tým ťažšie úlohy je schopná riešiť. V dnešnej dobe nie sme schopný vytvoriť nejakú univerzálnu neurónovú sieť, ktorá by bola schopná riešiť každý problém, ktorý jej predložíme. Pre každý problém, pre každú úlohu, máme samostatnú špecifickú neurónovú sieť. Tie sa líšia parametrami, funkciami a v mnohých iných aspektoch.

1 TEORETICKÁ ČASŤ ŠTUDENTSKEJ PRÁCE

1.1 Neurónová sieť

1.1.1 Inšpirácia

Umelé neurónové siete (ANN – Artificial Neural Network) sú inšpirované ľudským mozgom. Konkrétnejšie bunkami, zvanými neuróny. V ľudskom mozgu sa nachádzajú miliardy týchto buniek, ktoré sú prepojené ešte väčším počtom nervových vlákien, tzv. synapsii. Takýmto prepožovaním vzniká obrovská sieť, ktorá je zodpovedná za vzruchy, alebo javy, ako sú emócie, myšlienky a v konečnom dôsledku aj schopnosť učiť sa, vnímať a pohybovať sa [19].



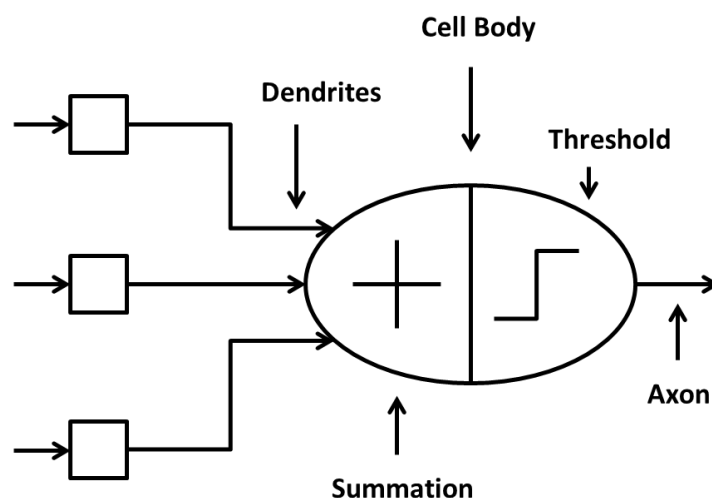
Obr. 1.1: Neurón

Každý neurón sa skladá z tela bunky, ktoré je centrom látkovej premeny a z mnohých výbežkov ktoré delíme podľa funkcie na:

- dendrity: sú centripetálne – prijímajú vstupnú informáciu (vzruchy z prostredia) [19]

- axóny (neurity) : sú centrifugálne – vedú informáciu v podobe nervového vzruchu von z tela neurónu [19]

Vo vnútri počítača je ekvivalentom k mozgovej bunke zariadenie zvané tranzistor. Základné typy mikroprocesorov obsahujú okolo 50 miliónov tranzistorov (vo veľkosti nanometrov). Nové technológie mikroprocesorov dokonca umožňujú natlačiť 2 miliardy tranzistorov na integrovaný obvod veľkosti 25mm² [5](pre predstavu, to je menej ako ma bežná poštová známka).



Obr. 1.2: Umelý neurón

1.1.2 Ako neurónové siete fungujú

Neurónové siete pristupujú k riešeniu problému inak ako bežné počítače. Tie používajú k riešeniu algoritmický prístup. Počítač postupuje podľa postupnosti príkazov a teda sa predpokladá, že vieme ako daný problém vyriešiť. Tieto príkazy sa prevádzajú do strojového kódu, ktorému počítač rozumie. Tieto stroje sú predvídateľné a preto pri nefunkčnosti vieme, že to je softwarová alebo hardwarová chyba. No pre nás by bolo výhodnejšie, keby počítače vedeli riešiť úlohy, pri ktorých nevieme, ako postupovať. ANN spracováva informácie podobne, ako ľudský mozog. Ako bolo už spomenuté, skladajú sa z obrovského počtu jednotiek, ktoré spracovávajú a následne riešia daný problém. Nedajú sa naprogramovať na riešenie špecifickej úlohy, pretože sa učia na základe príkladov, ukážok. Tieto ukážky musia byť zvolené s rozvahou,

pretože pri nevhodnom výbere môže prísť k nesprávnemu fungovaniu siete. Nedo-
kážeme predpovedať, ako bude postupovať ANN a preto pri nefunkčnosti nevieme,
kde nastala chyba. Klasické počítače a ANN medzi sebou nesúperia ale dopĺňajú sa.
Sú úlohy pre ktoré sa viac hodí algoritmickejší prístup a úlohy pre ktoré sa viac hodí
ANN. Pri veľmi veľkých počtoch úloh, sa uprednostňuje kombinácia oboch prístupov
(klasický počítač dohliada na ANN) pre maximalizáciu výpočtového výkonu [15].

1.1.3 Využitie neurónových sietí

Niektoré z využití ANN [2, 15]:

- rozpoznávanie a triedenie obrazov
- rozpoznávanie rukou písaného textu
- kompresiu obrázkov
- predpovedanie pohybov akcií na trhu
- medicína
- bezpečnosť

V poslednej dobe sa do popredia dostávajú najmä veľké spoločnosti ako sú Google,
Microsoft alebo Facebook.

Rozpoznávanie obrázkov

V roku 1980 chcel Pentagon využiť technológiu umelých neurónových sietí pri ochra-
ne tankov. Malo ísť o systém s kamerou, ktorá bude snímať okolie tanku a v prípade,
že spozoruje skrytý tank, pošle hlásenie posádke o prípadnom nebezpečenstve. Vývo-
jáři zobrali 100 snímok, na ktorých bol skrytý tank a 100 snímok na ktorých bolo
len prostredie (lesy, kríky a pod.). Z každej skupiny zobrali polovicu a použili ju pri
učení siete. Ako postupne prechádzala obrázkami a dávala odpovede na otázku, či ja
na fotke tank, vedci vždy kontrolovali výstup. Zo začiatku boli odpovede náhodné,
no časom sa zlepšila natoľko, že dokázala odhaliť tank. Aby si boli vývojári istí,
že sieť funguje ako má, zobrali druhú polovicu fotiek, s ktorými sa doposiaľ ANN
nestretla. Pri tomto veľkom teste tiež obstála. Fotky rozdelila podľa toho, či sa na
nich tank nachádza alebo nie. Pentagon bol s výsledkom spokojný, no chcel sieť pod-
robiť ešte jednému testu. Vytvorili úplne novú sadu záberov (na polovici boli tanky,
na druhej nie) a postupne nimi živili ANN. Odpovede boli zasa náhodné. Dlhú sa
nevedelo prísť na to, prečo tomu tak bolo, keď koniec koncov, ANN je „black box“
do ktorého nikto nevidí a teda nevieme, podľa čoho obrázky rozpoznáva. Neskôr
sa ukázalo, že Pentagon má niekoľko miliónový systém, ktorý vedel povedať, či je
slnečno alebo nie. Dôvodom tohto neúspechu bolo, že na prvotných 200 záberoch
boli tanky vyfotografované, keď bolo zamračené a snímky bez tankov za slnečného

dňa. Tu sa naplno ukázala nepredvídateľnosť umelých neurónových sietí. [7]

Na New York University (NYU) predstavili model pre generáciu obrázkov. Tento model sa neskladá z jednej, ale z dvoch neurónových sietí, ktoré spolu súperia [6, 11]. Prvá vytvára obrázky a snaží sa oklamať o druhú ANN o ich existencii. Druhá neurónová sieť sa snaží zistiť, ktoré obrázky sú reálne a ktoré sú zasa podvrh. Yann LecCun, vedúci oddelenia umelá inteligencia – Artificial Intelligence (AI) spoločnosti Facebook, to nazýva kontradiktórnym učením a zároveň verí, že to je cesta k obnoveniu poškodených fotografií. Výsledkom takéhoto systému môžu byť obrázky, u ktorých človek častokrát nemusí vedieť, či sa jedná o podvrh alebo nie.

Microsoft

Spoločnosť Microsoft má svoj vlastný projekt, o ktorom tvrdí, že je 2x spoľahlivejší a 50x rýchlejší ako všetky ANN doteraz [4]. Projekt s názvom Adam bol predstavený na 15-tom Microsoft Research Faculty Summit. Jednalo sa o mobilnú aplikáciu na rozoznávanie plemien psov. Pri predvádzaní mali na pódiu tri rôzne plemená psov a aplikácia v tomto prípade fungovala bezchybne. Pri predložení fotografie človeka reagovala slovami, že sa na fotke sa pes nenachádza [4]. Ďalej plánujú projekt rozšíriť aj o iné databázy obrázkov, ako napríklad kožných chorôb, alebo jedla, pre ľahké zistenie kalórií.

Microsoft taktiež experimentuje s myšlienkou prekladu reči v reálnom čase. V decembri 2014 uskutočnili, pomocou služby Skype, prvý oficiálny hovor v ktorom išlo o preklad z Angličtiny do Nemčiny a naopak. Preklad nebol dokonalý, no bol dostatočne dobrý na to, aby si dvaja ľudia rozumeli, bez akejkoľvek znalosti cudzieho jazyka. Keďže sa jedná o neurónovú sieť, ktorá sa stále učí, tak s každým ďalším prekladom sa úspešnosť prekladu zvyšuje.

Google

Spoločnosť Google sa venuje výskumu a tvorbe umelých neurónových sietí už zopár rokov. Za ten čas sa už zúčastnili mnohých súťaží, ktoré sú zamerané na triedenie obrázkov pomocou ANN. Pravidelne patria medzi elitu súťaží. Za zmienku určite stoja najmä MNIST, CIFAR a najviac zo všetkých ImageNet classification challenge, na ktorých ešte neurónové siete od Googlu neboli prekonané [16]. Napriek tomu, aké výsledky dosiahli, sa stále snažia prinášať nové spôsoby pre zrýchlenie a spresnenie neurónových sietí. Na to, aby niečo takéto dokázali, potrebovali pochopiť, ako presne funguje proces učenia, do ktorého nevidíme. Pri snahe zistiť túto skutočnosť sa im podarilo naučiť ANN vytvárať umelecké diela [11, 17]. Neurónová sieť hľadá na

obrázku známy obrazec a to opakuje stále dokola. Stále opakovanie tohto procesu v slučke, robí tento objekt ľahšie identifikovateľným, až kým ho na obrátku úplne nezobrazí. Napríklad, ak oblak vyzerá ako vták, sieť ho upraví natolko, že ako vták aj vyzerá.

Neurónové siete stoja aj za prepisom hovoreného slova. V tomto odvetví za posledné roky spoločnosť Google spravila tiež veľké pokroky. V roku 2009, keď sa začali prvýkrát zahrávať s myšlienkou prepisu hovoreného slova, používali metódu GMM (angl. Gaussian Mixture Model) [1, 3]. V roku 2012 viacvrstvové hlboké neurónové siete rozpoznávali zvuk lepšie ako GMM. Do podoby s akou sa stretávame dnes, napríklad v OS (angl. Operation System – operačnom systéme) Android, sa prepis dostal až vďaka príchodu RNN (Recurrent Neural Network – rekurentná neurónová sieť) a LSTM RNN (angl. Long Short-term Memory Recurrent Neural Network – rekurentná neurónová sieť s dlhodobou a krátkodobou pamäťou). LSTM RNN obsahujú okrem iného aj pamäť, do ktorej sa ukladajú dáta, na ktoré už sieť narazila. Ako keby interpretujeme slová, ktoré sme práve počuli slovami z inej vety.

Facebook

Facebook je ďalším z gigantov, ktorý sa začali zaujímať o ANN. Neurónová sieť, ktorú vytvárajú má nájsť použitie práve na ich sociálnej sieti. Má ísť o sieť, ktorá rozpozná objekty alebo ľudí na fotkách, text, poprípade bude označovať automaticky ľudí súvisiacich so statusmi a podobne. Ďalšou funkciou má byť cielená reklama. Facebook má prístup k obrovskému počtu dát, ktoré môže sieť spracovávať a síce kontá užívateľov [17]. Na základe toho, čo sa nám páči, chce Facebook v budúcnosti ponúkať reklamy, ako šité na mieru pre každého užívateľa.

Okrem iného výskumníci naučili svoju umelú inteligenciu vytvárať nové obrázky o rozmere 64x64 pixelov. Jedná sa o obrázky vecí ako sú automobily, lietadlá a zvierat. Podľa Roba Fergusa, výskumníka umelej inteligencie v spoločnosti Facebook, v 40% prípadov [11] sú dostatočne vierohodné na to, aby presvedčili človeka o svojej existencii.

Rozpoznávanie rukou písaného textu

Vytvorením bitovej mapy každého písmena a čísla sa chce zamedziť potrebe vytvorenia vlastného súboru znakov užívateľom.

Medicína

ANN sa používa pri diagnostike kardiopulmonálnych ochorení. V tomto prípade fungujú na základe porovnávania mnohých modelov. Dáta môžu obsahovať napr. krvný tlak, tep, frekvenciu dýchanie, taktiež údaje o pohlaví, veku a mnohé iné. Neurónová sieť sa z týchto informácií môže učiť a na základe týchto údajov vytvoriť obraz o zdravotnom stave jedinca. No predpokladom pre funkčnosť je pravidelnosť preventívnych kontrol u lekára [2].

ANN sa používa aj pre objektívnu diagnostiku sluchu u detí. Bežná metóda sa zakladá na reakcii mozgu na sluchový vnem, čo sa premietne na elektroencefalogram. Pri vyšetrení lekárom je treba test zopakovať aj viac ako 1000-krát [13], no pri použití neurónových sietí je možné prísť k výsledku už pri 200 opakovaníach, behom pár minút a to aj bez účasti lekára.

Bezpečnosť

V súčasnosti funguje systém Computer Aided Tracking and Characterization of Homicides (CATCH). Jedná sa o program, ktorý pomáha v dolapení sériových kriminálnikov. Známe dôkazy porovnáva s databázou iných prípadov, či už existujúcich, alebo vyriešených a hľadá medzi nimi podobnosti. Na základe podobností sa snaží odhadnúť ďalšie kroky páchateľa. Skladá sa z mnohých nástrojov, pri čom každý pracuje samostatne a každý dáva analytikovi iný pohľad na vec. Obsahuje nástroje pre spracovanie miesta činu, geografické mapy, nástroje pre vytvorenie charakteristiky obete a útočníka a mnohé iné [9].

1.1.4 Výhody a nevýhody umelých neurónových sietí

Výhody

- dosahujú najlepšie výsledky
- univerzálnosť
- zrýchlenie, zefektívnenie niektorých postupov, alebo operácií
- využiteľnosť v mnohých oblastiach ľudskej činnosti a ich zjednodušenie

Nevýhody

- black box (nemáme predstavu na základe čoho sa sieť učí)
- hardwerová náročnosť
- pre vytvorenie ANN je potrebná znalosť neurónových sietí

1.2 Učenie neurónových sietí

Jednou z vlastností pre ktorú, sú neurónové siete také špecifické je schopnosť učiť sa a tým zlepšovať výsledok. K zlepšeniu prichádza postupne. ANN, podobne ako človek, sa učí interakciou s prostredím a následne spätnou väzbou. Informácie sú privádzané na vstup, čo spôsobí spustenie vrstiev na ktorých sú jednotky, ktoré informáciu prenášajú až na výstup. Pri prenose informácie sa neaktivujú však všetky tieto jednotky, ale len tie ktoré sú na základe váhy (dôležitosti) nutné. V ideálnom prípade získava nové znalosti pri každej jednej interakcii s prostredím [12, 13, 18].

Pri neurónových sieťach je obtiažne definovať pojem učenie, pretože sa s ním spája až príliš veľa aktivít a samozrejme záleží aj na uhle pohľadu. Učenie je proces, pri ktorom sú nastavené parametre umelej neurónovej siete v závislosti na prostredí, s ktorým prichádza do kontaktu.

Predpokladá sa nasledovné:

1. Neurónová sieť dostáva podnety z okolia.
2. V dôsledku bodu 1, sa menia parametre siete.
3. Po zmene parametrov sieť reaguje na podnety v zmenenej forme, teda inak.

Vyššie uvedené body môžeme nazývať algoritmom učenia, no zároveň treba brať do úvahy, že existuje mnoho obdobných algoritmov a každý je vhodný pre iný druh neurónovej siete. Algoritmy sa líšia napríklad spôsobom, akým nastavujú váhu (dôležitosť) jednotlivým synapsiám, alebo spôsobom komunikácie s prostredím v ktorom pôsob ANN pôsobí.

1.2.1 Postup učenia

- Zbieranie dát.
- Príprava a normalizácia dát.
- Výber topológie siete.
- Prvotné nastavenia
- Proces učenia
- Kontrola správnosti učenia
- Prispôbovanie parametrov, záverečné učenie.
- Vizualizácia siete pre ďalšie použitie.

Zbieranie dát

Výber dát, ktoré sa použijú pri učení siete, je jedna z najťažších fáz a preto musia spĺňať niekoľko kritérií:

- Reprezentatívnosť – údaje musia odrážať skutočný stav v danej oblasti
- Konzistencia – malo by sa zamedziť výskytu protikladov, čo by mohlo viesť k nekvalitnému učeniu siete.

Dáta sa následne upravujú do formátu vhodnom pre vstup do neurónovej siete. Jeden záznam v databáze sa nazýva vektor učenia. Vektor obsahuje hodnotu pre každý vstup siete a taktiež jednu hodnotu pre každý výstup siete, bez ohľadu na to, či sa jedná o učenie s učiteľom, alebo bez neho. Proces 'vnímania' siete, sa dá ešte vylepšiť niekoľkými spôsobmi:

- Normalizácia: upravovanie rozmernosti všetkých vstupných a výstupných dát na rovnakú úroveň
- Kvantizácia: sa využíva pri nepretržitých premenných, pre ktoré existuje konečné množstvo diskretných hodnôt. Používa sa napríklad pri nastavovaní frekvencie zvukových signálov pri rozpoznávaní reči.

Dôležitý je taktiež výber topológie siete. Čím viac výstupov sieť má, tým markantnejší je rozdiel medzi triedami a v konečnom dôsledku to zníži chybovosť siete.

Výber topológie

Topológiu treba vyberať s ohľadom na úlohu a dáta ktoré máme k dispozícii. Najčastejšie sa vyskytujú tieto topológie:

- Feedforward: architektúry vhodné pre spracovanie veľkého množstva dát z pevných pamätí. Sú to viacvrstvové neurónové siete, ktoré neobsahujú žiadne slučky ani cyklenie. Informácie sa pohybujú jedným smerom, od vstupnej vrstvy, cez skryté vrstvy (môže byť jedna, alebo viacej) až po výstupnú vrstvu.
- Cellular: Pri tomto druhu neurónových sietí môžu bunky komunikovať len so susednými bunkami. Obsahujú viacero vstupov no len jeden výstup. Sú vhodné pre spracovávanie obrázkov, 3D analýzu povrchov a iné.
- Feedback: Alebo inak cyklické neurónové siete. V tejto architektúre bunky tvoria uzavretý kruh, čo im umožňuje vytvoriť modely, alebo riadiť dynamické systémy.

Prvotné nastavenia

Prvotné nastavenia závisia od druhu siete a jej zamerania. Pri niektorých môže ísť o počet vrstiev, pri iných o počet buniek skrytých vrstiev. Pri výbere počtov vrstiev a neurónov treba brať do úvahy následný dopad na funkciu siete. So vzrastajúcim počtom prepojení medzi bunkami sa zvyšuje schopnosť siete zovšeobecňovať. Počet týchto prepojení je závislý na počte dát, ktoré sú určené k učeniu siete. Toto bolo v stručnosti nastavenie vlastností.

Ďalšou dôležitou vecou, je nastavenie parametrov. Toto sa týka predovšetkým sietí, ktoré sa učia s učiteľom. Na správnej voľbe parametrov závisí rýchlosť a presnosť učenia. Preto sa nastavujú s ohľadom na kritéria učenia. Ak chceme aby prebehlo rýchlo, nastavíme vyššiu rýchlosť učenia, čo môže spôsobiť rast chybovosti. Naproti tomu pomalšie učenie spôsobí zvýšenie doby konvergenencie odpovede, ale môžeme sa tak vyhnúť nefunkčnosti siete. Preto sa treba už pri návrhu rozhodnúť, či chceme minimalizovať chyby, alebo čas potrebný k učeniu siete.

Proces učenia

Pri procese učenia neurónová sieť prechádza dátami, ktoré má k dispozícii. Tento prechod môže byť náhodný, alebo systematický. Môže každým prvkom prejsť raz, alebo viackrát, v závislosti na type ANN. Pokiaľ prechádza dátami viac ako raz, tak jeden celý prechod sa nazýva epoch učenia. Pri učení s učiteľom delíme dáta na:

- súbor dát určených k učeniu
- súbor dát určených k testovaniu

Princíp delenia dát môže byť ľubovoľný. Dáta určené k učeniu sa privádzajú na vstup, aby sa na nich sieť učila a dáta určené k testovaniu sa používajú pri výpočte chýb. Tieto dáta sa nikdy nepoužívajú k učeniu. Môžu nastať dve situácie:

1. Chyba na testovacích dátach sa znižuje, čo znamená, že umelá sieť úspešne zovšeobecňuje.
2. Chyba dát určených k učeniu klesá, no chyba na testovacích dátach stúpa, znamená to, že sieť nezovšeobecňuje, ale len si pamätá učebné dáta a spôsob, akým má na ne reagovať. V takomto prípade hovoríme o preučení (overfitting-u) siete a učenie sa zväčša zastaví.

V priebehu učenia sa môžu vyskytnúť mnohé chyby, ktoré sa nedajú predpovedať a zároveň sa na ne nedá vhodne reagovať, pretože umelé neurónové siete sú vlastne

čierne škatule, do ktorých nikto poriadne nevidí.

Kontrola správnosti učenia

Na prvý pohľad úspešné učenie nie je garanciou úspechu. Nikdy totiž nevieme, čo sme sieť naučili, ako sa napríklad stalo s neurónovou sieťou pre rozpoznávanie tankov na obrázku. Testovanie sa prevádza na príkladoch, ktoré neboli zapojené do učenia a príkladov musí byť o to viac, čím bolo učenie kvalitnejšie. Napríklad pre neurónovú sieť s pravdepodobnosťou chyby blízkej k milióntine je potrebné pri testovaní použiť milión príkladov. Čím kvalitnejšie je sieť učená, tým zložitejšia je kontrola správnosti učenia.

1.2.2 Modely učenia

Existujú tri významnejšie spôsoby učenia, pri čom každý je charakterizovaný inými metódami.

1. Učenie s učiteľom
2. Učenie bez učiteľa
3. Posilnené učenie

Pri učení s učiteľom sú známe správne odpovede pre vstupné dáta a následne sa upravujú váhy pre minimalizovanie chýb. Pri metóde bez učiteľa rozdeľuje vzory na základe vnútornej štruktúry. V tretej metóde ide o prístup softwerového agenta na podnety z prostredia.

Učenie s učiteľom

Z anglického *Supervised learning* (učenie pod dohľadom) – využíva k učeniu dvojicu parametrov, podnet a odozvu [12, 13, 18]. Medzi vstupom (podnet) a výstupom (odozva) existuje nejaká závislosť, ktorá je však neznámá. Na základe konečnej množiny dvojíc sa snaží znova postaviť model vzťahov medzi podnetom a odozvou. V konečnom dôsledku ide o zostavenie algoritmu, ktorý je schopný pre ľubovoľný podnet reagovať vhodnou odpoveďou.

Systém pozostáva zo systému podnetov z prostredia a z regulátoru vnútorných parametrov. Ako regulátor sa dajú použiť automatické regulačné zariadenia, ako sú termostaty, alebo operátor (človek – učiteľ), ktorý je schopný reagovať na odozvy a podnety na základe pravidiel, čím príde ku zmene stavu pamäti v systéme.

Typy vstupných dát:

- Atribútový popis – každý objekt má svoje vlastnosti (atribúty), ktorými je popísaný. Tieto vlastnosti môžu byť číselné alebo nečíselné. Používa sa najčastejšie.
- Matice vzdialeností medzi jednotlivými objektmi. Objekty sú popísané vzdialenosťami od všetkých ostatných objektov v učebnej množine. Tento typ vstupu využívajú len niektoré metódy, predovšetkým metódy najbližšieho suseda a potenciálových funkcií.
- Pri signáloch môže byť meranie reprezentované vektorom, číslom, alebo vo všeobecnosti atribútami v určitom čase.
- Videá a obrázky.
- V zložitejších prípadoch sa môžu ako vstupy použiť aj grafy, texty, alebo výsledky dotazníkov, no z pravidla je nutné previesť tieto vstupy do formy atribútov.

Rozlišujeme tri typy odozvy:

- Úlohy regresie a aproximácie: nastávajú v prípade, že množina možných odpovedí je nekonečná.
- Úlohy klasifikácie a rozpoznávania vzorov: jedná sa o konečnú množinu odpovedí.
- Úlohy prognózy: odpovede predpovedajú chovanie javov, alebo procesov.

Učenie bez učiteľa

Z anglického *Unsupervised learning* (učenie bez dohľadu, samoštúdium) – metóda pri ktorej sa stroj učí bez zásahu človeka [12, 13, 18]. Metóda je vhodná len pri úlohách v ktorých ide o nájdenie vzťahov, závislostí medzi objektami. Pri tejto metóde sa počíta s popisom množiny učebnej sady.

Pri tvorbe neurónových sietí nám ide o to, aby sa funkciou čo najvernejšie podo-

bali ľudskému mozgu. No v tom sa nestretneme s mechanizmom, ktorý by porovnával výstupy s požadovanými výstupmi. Preto sa skupina vedcov začala zaoberať problematikou 'poludštenia' systému. Preto metóda vytvorená Kohonenom nepotrebuje cieľový vektor a preto sa u nej nevyskytuje porovnávanie s nejakými ideálnymi odpoveďami.

Učenie bez učiteľa sa najčastejšie používa pri rozpoznávaní obrazov. V závislosti na prístupe sa formuje matematický koncept. Ide o identifikovanie spoločných vlastností pri tvorbe vhodného popisu, za použitia známych informácií. Podľa prístupu rozlišujeme tri metódy:

- Matematické metódy.
- Syntaktické metódy.
- Heuristické metódy.

Na vstup sa privádzajú informácie vo forme matíc, ktorý je následne rozdelení do disjunktných podmnožín – zhlukov – tak aby každá podmnožina obsahovala podobné objekty, no zároveň, aby sa výrazne líšila od iných podmnožín. Súbor dát je potrebné rozčleniť do zhlukov, následne pre každý aplikovať nejakú jednoduchú metódu, pri čom tento proces môže hrať úlohu pri riešení problémov. Niektoré z metód riešenia:

- Kohenenova neurónová sieť.
- Hierarchické zhľukovanie.
- Grafové algoritmy zhľukovania.
- Štatistické algoritmy zhľukovania.
- Neurónová sieť spätného šírenia.

Pri hľadaní vlastností medzi dátami, je úlohou stroja nájsť funkcie a hodnoty atribútov, ktoré sa medzi dátami objavujú často, za predpokladu, že vstupné dáta obsahovali atribútový popis. Počet atribútov môže byť obrovský, preto je potrebná

prezentácia dát v nižších dimenziách a minimalizácia stratovosti dát. Toto zabezpečujeme:

- Metódou hlavných komponentov.
- Metódou nezávislých komponentov.
- Viacrozmerným stupňovaním.

Učenie s učiteľom sa používa najmä pri kategorizácii v mnohých odvetviach, ako napríklad pri sociologických výskumoch, v marketingu pri identifikácii preferencií a podobne.

Posilnené učenie

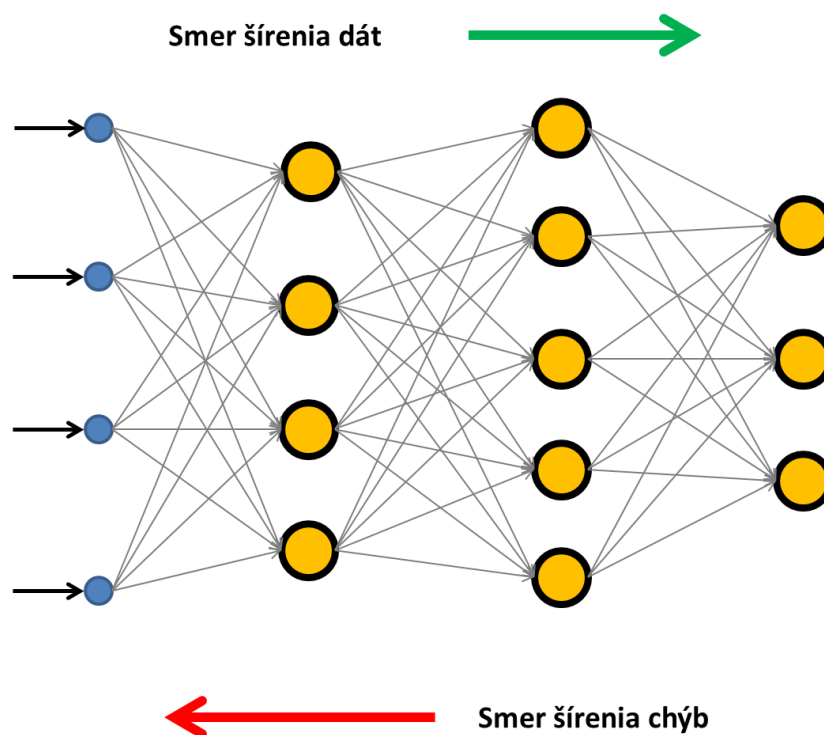
Z anglického *Reinforcement learning* – je metóda pri ktorej sa stroj učí z interakcií s prostredím [12, 13, 18]. Jedná sa o systém so spätnou väzbou, teda agent (stroj) pôsobí na prostredie a naopak prostredie pôsobí na agenta. Odozva prostredia je vlastne podporným systémom pre učenie. Keďže sieť obsahuje podporné systémy, môžeme ho považovať za istý druh učenia s učiteľom (učiteľom je prostredie). No môžeme ho zaradiť aj do učenia bez učiteľa a to na základe implicitných učiteľov v prípade umelého neurónového prostredia.

Systém podpory učenia je súbor pravidiel, u ktorých je možné časom meniť stav pamätí, poprípade matice interakcií. Okrem metódy učenia s učiteľom, Rosenblatt [14] zaviedol pojem učenie bez učiteľa a vytvoril niekoľko spôsobov učenia.

- Alfa systémy: váha spojení vedúcich k prvku sa všade mení o tú istú hodnotu a váha neaktívnych spojení sa nemení.
- Gama systémy: ide o pravidlo zmeny váh v systéme. Váha všetkých aktívnych spojení sa nastaví na tú istú hodnotu a následne je odčítaná z každého spojenia iná hodnota, ktorá sa rovná úplnej zmene váh aktívnych spojení, vydelených celkovým počtom spojení. Plná suma váh spojení sa nemôže ani zvýšiť ani znížiť.

Pre každý problém existuje algoritmus učenia. Jedným z najlepších algoritmov je už popísaný rekurentný algoritmus (algoritmus spätného šírenia). Myšlienkou je zmena váh synapsii na základe chybovej funkcie. Rozdiel medzi odpoveďami siete a očakávanými odpoveďami sa určuje na výstupnej vrstve a šíri sa v opačnom smere, na

základe čoho je schopný každý neurón určiť svoj prínos k celkovej chybe. Nevýhodou takéhoto algoritmu je neistota správnosti učenia.



Obr. 1.3: Smer toku dát a chýb v RNN

Prínos grafických procesorov

Za mozog počítačov sa dlhé roky považoval procesor. Ten vykonával všetky aritmetické operácie. Jeho doplnení boli čipy, ktoré zobrazovali veci na monitoroch. No v dnešnej dobe sa stále viac rozpráva o grafických procesoroch (GPU – Graphics Processing Unit), ktoré nám vďaka odlišnej architektúre umožňujú využívať oveľa viac jadier, ako bežné procesory (CPU – Central Processing Unit). Zo začiatku sa GPU využívalo na vykresľovanie 3D obrazov, no stále viac nachádzajú využitie v oblasti výpočtov. Využívajú sa pri revolučných vedeckých objavoch, pri tvorbe modelov vo finančníctve a v mnohých iných odvetviach.

Nathan Brookwood opísal jedinečné vlastnosti grafických procesorov týmito slovami: *‘Grafické procesory sú optimalizované tak, aby mohli zobrať obrovské zhluky dát a opakovali stále dokola tú istú operáciu vo veľmi veľkej rýchlosti. Narozdiel od počítačových mikroprocesorov, ktoré by medzi týmito dátami skákali.’* [10]

Architektúra CPU bežne obsahuje pár výpočetných jadier a rýchle vyrovnávacie pamäte, ktoré sú schopné naraz zvládnuť len malý počet softvérových vlákien. Naproti tomu máme GPU, ktoré sa skladajú zo stovky jadier, ktoré dokážu zvládnuť naraz až tisícky výpočetných vlákien a tým výpočty urýchliť. Napríklad GPU so 100+ jadrami dokáže v niektorých prípadoch rýchlosť softvéru zvýšiť až 100x oproti bežnému CPU [10]. Okrem iného sú elektricky menej náročné a lacnejšie, no hodia sa len na istý typ výpočetných úloh. Práve to sú hlavné dôvody, prečo sa grafické procesory začali využívať pri učení neurónových sietí. Stále však treba myslieť na to, že najlepšie výsledky dosahujú kombinácie oboch procesorov, ako grafického, tak aj bežného počítačového.

1.3 Caffe

1.3.1 Úvod do Caffe

Caffe je systém pre hlboké učenie, ktorý vyvíja Berkeley Vision and Learning Center (BVLC) v spolupráci s prispievateľmi zo širokej verejnosti. Jedná sa o opensource systém, ktorého tvorcom je *Yangqing Jia*. Výhody Caffe [8]:

- Expresívna architektúra: Jednoduchosť pri tvorbe modelu a optimalizácii, bez zložitého programovania. Možnosť prepínať medzi CPU a GPU pomocou jediného indikátora. Preto je jednoduché neurónovú sieť učiť cez GPU a následne

ju využiť napríklad v mobilných zariadeniach.

- Možnosť úprav: Keďže sa jedná o opensource systém, ktorý bol už za prvý rok existencie použitý viac ako tisíckou vývojárov, prešiel mnohými zmenami a stále užívatelia prispievajú novými postrehmi a nápadmi. Aj vďaka tomu patrí Caffé k vrcholu čo sa týka kódu aj samotného modelu.
- Rýchlosť: Pri použití kvalitného grafického procesora, dokáže model Caffé spracovať až 60 miliónov obrázkov za deň, čím pripadá 5 ms/obrázok. Touto rýchlosťou patrí medzi tri najrýchlejšie deep-learning systémy na svete.
- Prístupnosť: Vďaka tomu, že je voľne dostupný, rýchly a jednoduchý na obsluhu, je súčasťou už mnohých projektov, od prototypov až po aplikácie veľkých rozmerov, akými sú rozpoznávanie reči a obrazov.

1.3.2 Prehľad vrstiev

Zovšeobecnení môžeme povedať, že neurónová sieť sa skladá z troch vrstiev:

1. vstupnej
2. výstupnej
3. skrytej

Teraz nás zaujíma práve tretia spomínaná, skrytá vrstva. Tá sa môže skladať z jednej alebo viacerých vrstiev, o čom rozhoduje tvorca siete. Tieto vrstvy sa líšia funkciou a logicky parametrami, ktoré pri nich nastavujeme. Najzákladnejšími typmi skrytých vrstiev sú [8]:

- Convolution layer: Jedná sa o vrstvu, ktorá prechádza obrázok kernelom (konvolučnou maticou) . V sieti sa môže nachádzať viacero takýchto vrstiev, pričom každá má zadanú vlastnú maticu, ktorou prechádza postupne celý obrázok, pričom parametre matice sa nemenia. Úlohou týchto vrstiev je vytiahnuť z obrazov niečo, čo sa líši od zvyšku obrazu. Prvá vrstva zväčša odhalí najprimitívnejšie črty, ako sú rohy, krivky alebo obrysy. Z čím väčšieho počtu konvolučných vrstiev sa ANN skladá, tým detailnejšie, zložitejšie rysy si začne všímať.
- ReLu layer: Rectified Linear Unit – je vrstva, ktorá využíva aktivačnú funkciu $f(x)=\max(0,x)$, ktorá je bez prírastku. Zvyšuje nelinearitu rozhodovacej

funkcie, ale aj siete samotnej, bez toho, aby nejakým spôsobom ovplyvňovala konvolučnú vrstvu. Na zvýšenie nelinearity sa môžu použiť aj funkcia hyperbolického tangensu, no pri jej použití sa sieť učí niekoľkonásobne pomalšie v porovnaní s ReLU.

- Pooling layer: Aby sa zamedzilo zmene obrazu po prechode konvolúciou, táto vrstva vypočítava maximálnu, alebo priemernú hodnotu špecifickej črty v častiach obrazu. Tento prvok je nesmierne dôležitý najmä pri klasifikácii obrazov, alebo pri detekcii prvkov na obrázkoch.
- Dropout method: Jedná sa metódu ktorá má zabráňovať zahlteniu siete, keďže úplne prepojené vrstvy zaberajú väčšinu parametrov. Pri každom procese učenia sú jednotlivé uzly buď zahodené, alebo ponechané. Ponechaná je len zredukovaná sieť, pri čom sa zahadzujú aj vetvy, ktoré z uzlu odchádzajú, alebo do uzla vchádzajú. Takto pretvorená sieť pokračuje v procese učenia. Odstránené uzly sú následne vložené späť do siete, s parametrami, ktoré mali nastavené pred začiatkom učenia.
- Loss layer: Sú určené na minimalizáciu chyby. Porovnávajú výstup s očakávanou odpoveďou. V prípade chyby sa snažia túto chybu minimalizovať. Vďaka tomu, že postupujú proti smeru toku dát, vedia určiť, v ktorom neuróne nastali chyby a vedia ich následne korigovať. Existuje viacero typov funkcie loss, na základe problémov ktoré sieť rieši.
- LRN layer: Local Respose Normalization – jedná sa o druh normalizácie na lokálnych vstupoch.

2 VÝSLEDKY STUDENTSKÉ PRÁCE

Táto kapitola obsahuje stručný úvod do problému a popis fungovania programu, ako aj popis jednotlivých tried.

2.1 Úvod do problému

V našom prípade sa budeme zaoberať neurónovými sieťami, ktoré majú čo dočinenia s Caffe. Ako už bolo spomínané, Caffe je systém pre hlboké učenie, ktorý vyvíja BVLC v spolupráci s prispievateľmi zo širokej verejnosti. K definícii neurónových sietí využíva textový dokument „prototxt“. V tomto textovom dokumente sa nachádzajú všetky vrstvy a ich počiatočné nastavenia a je charakteristický svojou štruktúrou. Štruktúra je zhodná so štruktúrou, ktorá je známa pod názvom **ProtocolBuffers**. Tá je vyvíjaná spoločnosťou Google a slúži pre spracovávanie dátových štruktúr. Je menšia ako XML, rýchlejšia a jednoduchšia. Dá sa upravovať podľa potreby a je podporovaná mnohými platformami (C++, Java, Python). V našom prípade sa na delenie textu využíva fakt, že každá vrstva sa začína postupnosťou znakov „layer {“.

Úlohou semestrálneho projektu bolo vytvoriť program, ktorý takýto textový súbor načíta a následne rozdelí na jednotlivé vrstvy. Tieto vrstvy následne zobrazí, teda vytvorí grafické rozhranie neurónovej siete.

2.2 Tvorba programu

Pri tvorbe programu sa používal programovací jazyk **Java**. Konkrétnejšie balíček **JavaFX**, ktorý je vhodný pre tvorbu grafického rozhrania a zároveň je modernejší ako **Swing**. Používala sa verzia **JDK 8**. Pri práci sa použil aj **Maven**, pre správnu implementáciu knižnice **ProtocolBuffers**. Program sa skladá z viacerých tried, pričom každá sa stará o niečo iné. Program načíta textový dokument, ktorý sa následne rozdelí na menšie časti, vrstvy, ktoré sa potom zobrazia a je možné si ich po jednom prehliadať. Funkcia programu v bodoch:

1. Načítanie súboru.
2. Rozdelenie textu zo súboru.
3. Vytvorenie tried.
4. Vizualizácia neurónovej siete.

```
1 name: "FlickrStyleCaffeNet"
2 layer {
3   name: "data"
4   type: "ImageData"
5   top: "data"
6   top: "label"
7   include {
8     phase: TRAIN
9   }
10  transform_param {
11    mirror: true
12    crop_size: 227
13    mean_file: "data/ilsvrc12/imagenet_mean.binaryproto"
14  }
15  image_data_param {
16    source: "data/flickr_style/train.txt"
17    batch_size: 50
18    new_height: 256
19    new_width: 256
20  }
21 }
22 layer {
23   name: "data"
24   type: "ImageData"
```

Obr. 2.1: Ukážka súboru prototxt

2.2.1 Loader.java

Táto trieda sa stará o načítanie, rozdelenie textu a jeho uloženie, pre ďalšie používanie. Vstupným parametrom triedy je `String` ktorý tvorí cieľová adresa súboru. Tá sa najskôr zadávala staticky, no neskôr sa prešlo na získavanie adresy pomocou metódy `FileChooser`. Tým sa zabezpečilo dynamické načítavanie súborov. Po načítaní textového súboru sa pristupuje k deleniu textu. Zo začiatku sa text delil len príkazom `split` a následne sa jednotlivé časti ukladali do `ArrayListu`. Toto sa časom ukázalo ako nepostačujúce, respektíve nevhodné riešenie, vzhľadom na neskoršiu potrebu získavania určitých údajov z jednotlivých vrstiev.

Preto sa pristúpilo k inému riešeniu problému. Na oficiálnych stránkach pre Caffe sa nachádza súbor `caffe.proto`. Tento súbor obsahuje definície všetkých vrstiev, podľa ktorých sa vytvorí Java trieda. K vytvoreniu tejto triedy sa používa ešte program `protoc.exe`. V priečinku, ktorý obsahuje tieto dva komponenty, sa otvorí príkazový riadok a cezeň sa spustí program `protoc.exe` s cestou k súboru `caffe.proto`. Týmto spôsobom sa vytvoril už spomínaný balíček s vrstvami a funkciami pre jednoduchú prácu s nimi. Každá vrstva obsahuje tieto parametre:

- Name - meno
- Type - typ (Data, ReLu, Convolution, ...)
- Top - názov vrstvy pre ktorú je daná vrstva nadradená
- Bottom - názov vrstvy, z ktorej daná vrstva dedí

Okrem týchto, môže vrstva obsahovať množstvo iných parametrov, ktoré sú pre každú vrstvu špecifické. Samozrejme aj tu sa nachádzajú výnimky, ako napríklad vrstvy s dvoma, alebo viacerými Bottom parametrami, čo sa muselo taktiež ošetriť. Tento proces funguje na základe už spomínaného balíčka `ProtocolBuffers`.

V ďalšom kroku sa pristupuje k vytvoreniu „ArrayListu“ do ktorého sa ukladá text rozdelený na jednotlivé vrstvy. Delenie textu sa robí v niekoľkých krokoch, opakujúcich sa vo `for` cykle.

1. Vytvorenie „buildera“ pomocou:
`caffe.Caffe.LayerParameter.newBuilder().`
2. Úprava už rozdeleného textu odstránením prebytočných zátvoriek na začiatku a konci textu (nepovinné).
3. Vytvorenie vrstvy pomocou „buildera“ a následné pridanie do „ArrayListu“.

Ako výstup z triedy sa využíva vzniknutý `ArrayList`.

2.2.2 LayerVis.java

Tato trieda je vytvorená za účelom vizualizácie vrstiev. Ako vstupné parametre preberá vrstvu, poradie a počet vrstiev, ktorý udáva, koľko vrstiev z danej vrstvy dedí. V tejto triede sa nastavujú parametre pre vizualizáciu jednotlivých vrstiev. Pre vizualizáciu vrstiev sa zvolili kruhy, ktoré sú farebne rozlíšené podľa typu vrstvy. Trieda obsahuje podmienky, ktoré sa starajú o posun súradnice stredu na osi *x* v prípade, že z vrstvy dedí viacero vrstiev. Takýmto spôsobom vzniká odsadenie, ktoré zabezpečuje lepšiu prehľadnosť. Súradnica stredu na ose *y* sa posúva na základe parametru poradie, ktorý sa s každou ďalšou vrstvou inkrementuje. Taktiež obsahuje metódy pre uchovanie názvu danej vrstvy a súradníc stredu kruhu. Pre priradenie farby sa používa `HashMap`, do ktorej sa staticky zadali potrebné údaje, teda typ vrstvy a farba, ktorá ju má reprezentovať. Táto trieda volá triedu pre tvorbu kruhu, ktorej odovzdáva súradnice pre stred, polomer a farbu. Ďalej obsahuje premennú, do ktorej sa ukladá text s parametrami, ktorá sa používa pre navrátenie vrstvi do pôvodného stavu.

2.2.3 Cir.java

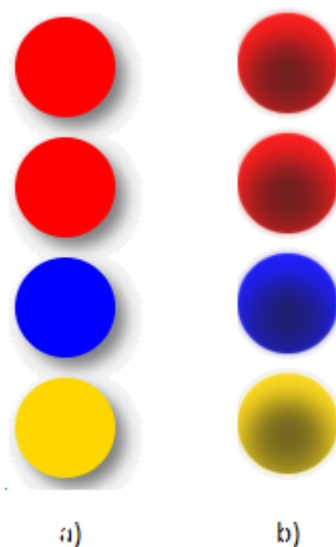
Je trieda, ktorá sa využíva pre tvorbu kruhu a následnú úpravu dizajnu. Tu sa vyskúšalo viacero možností. Najskôr sa použila možnosť, kedy sa vytvoril kruh s tieňom a svetlejší kruh. Následne sa nimi prekryl už existujúci kruh, no bolo veľmi náročné nájsť optimálnu polohu a polomer, aby sa docielil očakávaný výsledok. Preto sa prešlo na možnosť, pri ktorej sa používa `javafx.scene.effect.DropShadow`, ktorému sme nastavili sivú farbu.

2.2.4 GUI.java

Jedná sa o hlavnú triedu, ktorá obsahuje metódy `main` a `start`. V metóde `start` sa vytvára hlavná scéna, pre ktorú sa využíva rozloženie `BorderPane`. Toto rozloženie rozdelí scénu na vrchnú, spodnú, ľavú, pravú a strednú časť. V našom prípade sa používa len vrchná, stredná a pravá časť.

Do vrchnej časti bolo pridané jednoduché menu, obsahujúce možnosti pre otvorenie a uloženie súboru, ako aj možnosť pre ukončenie programu. Načítanie a ukladanie sa vykonáva pomocou metódy `FileChooser`. Pri načítavaní súboru sa ukladá cesta k súboru, ktorá sa neskôr používa ako parameter pri volaní triedy `Loader.java`. Pri ukladaní sa vytvorí `StringBuilder` do ktorého sa na začiatok uloží názov súboru a potom sa pomocou cyklu `for` rozširuje postupne o text z jednotlivých vrstiev. Ukončenie programu sa vykonáva príkazom `System.exit(0)`.

V pravej časti sa taktiež aplikuje rozloženie `BorderPane`, pričom do spodnej časti

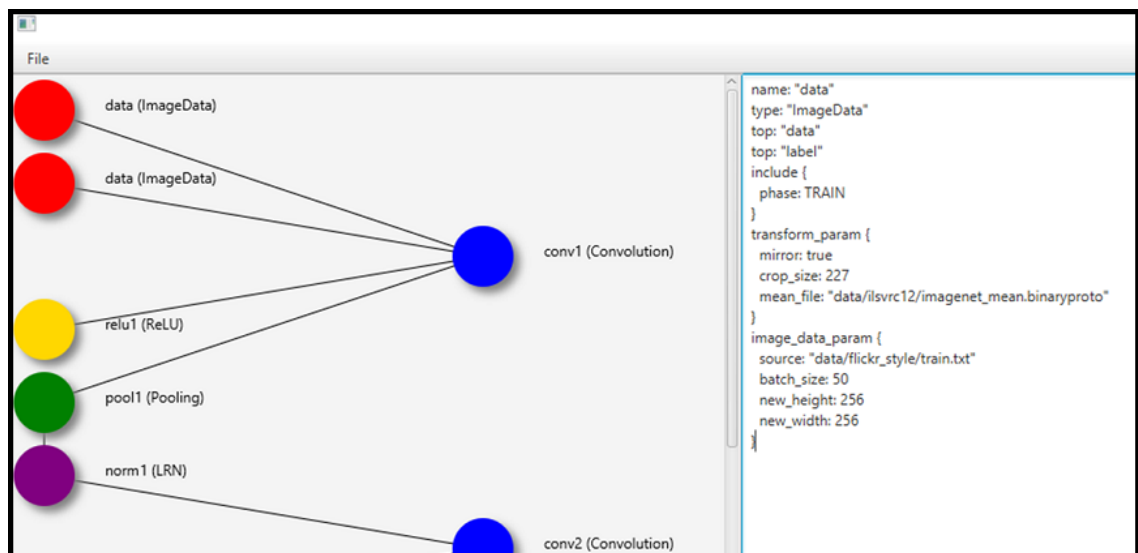


Obr. 2.2: a) riešené tmavým a svetlým kruhom , b) DropShadow

je umiestené tlačidlo pre navrátenie vrstvy do pôvodného stavu. V centrálnej časti je umiestená plocha pre výpis textu vrstvy. Text sa vypisuje po kliknutí na kruh, reprezentujúci vrstvu.

V centrálnej časti sa aplikuje **ScrollPane**, do ktorého sa vkladajú jednotlivé kruhy a čiary. Vizualizácia sa vykonáva pomocou metódy, ktorá sa skladá z vnorených **for** cyklov. Tie slúžia pre počítanie vrstiev, ktoré dedia z tej istej vrstvy. Následne sa volá už spomínaná trieda **LayerVis.java**, ktorej výstup sa ukladá do **ArrayListu**, pre ďalšie používanie. Pre funkciu klikania sa pridala funkcia **MouseEvent**. Napravo od kruhu sa pomocou metódy **showInfo** zobrazuje názov a typ danej vrstvy.

Ďalej táto trieda obsahuje metódu pre vytvorenie grafického spojenia vrstiev, medzi ktorými existuje hierarchické spojenie. Pre spojovanie vrstiev sa využila metóda **Line**, ktorej sa odovzdávajú počiatočné a koncové súradnice, ktoré sú v našom prípade reprezentované súradnicami stredou dvoch kružníc, reprezentujúcich vrstvy. Pre vykresľovanie čiar sa používa ďalší vnorený cyklus **for**, ktorý porovnáva „Bottom“ parameter každej vrstvy, s názvom danej vrstvy. Ak sa tieto dva parametre zhodujú, vytvorí medzi oboma vrstvami čiaru. Súradnice pre počiatok a pre koniec sa vezmú z **ArrayListu** v ktorom sú uložené výstupy z triedy **LayerVis.java**. Trieda obsahuje aj metódu pre už spomínaný návrat vrstvy do pôvodnej podoby.



Obr. 2.3: Ukážka programu

3 ZÁVER

Úlohou práce bolo vytvoriť program pre vizualizáciu umelých neurónových sietí. Práca obsahuje okrem postupu tvorby programu aj stručnú analýzu umelých neurónových sietí. V analýze sa rieši ich funkcia, učenie, využitie, ale aj problémy s ktorými sa najčastejšie stretávame. Sú v nej uvedené spoločnosti, ktoré sa momentálne najviac zaujímajú o rozvoj tejto technológie, ako aj ich úspechy. Čím sa analýza nezaoberá, sú teórie o nebezpečenstve, ktoré nám hrozí pri ďalšom vývoji umelých neurónových sietí.

Vytvorený program môžeme rozdeliť na dve časti. Prvá, pri ktorej tvorbe programu sa využíval programovací jazyk **Java**, sa stará o načítavanie textu zo súboru a jeho rozdelenie na jednotlivé vrstvy. Pri tvorbe tejto časti sa používala knižnica **ProtocolBuffer** a súbory z oficiálnej stránky tvorcov **Caffe**. Knižnica spolu so súbormi sa využila pri tvorbe triedy, ktorá uľahčila ukladanie jednotlivých vrstiev, ako aj rozširovanie funkcií programu v budúcnosti.

Druhá časť, pri tvorbe ktorej sa používala **JavaFX**, sa stará o samotnú vizualizáciu. Pre jednotlivé vrstvy vykreslí kruhy, ktoré sú farebne rozdelené podľa typu vrstvy a majú vedľa seba popis s názvom danej vrstvy. Pokiaľ medzi niektorými vrstvami existuje hierarchia, tieto vrstvy spojí čiarou. Vytvorené prostredie taktiež obsahuje časť, do ktorej sa vypíšu parametre vrstvy. Táto funkcia sa aktivuje kliknutím na názov, alebo na kruh reprezentujúci vrstvu a tlačidlo pre zvrátenie zmien.

LITERATÚRA

- [1] Andrew Senior, F. o. B., Kanishka Rao: Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition.
URL <http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43894.pdf>
- [2] Application of neural networks. 2000.
URL <http://cs.stanford.edu/people/eroberts/courses/soco/projects/2000-01/neural-networks/Applications/index.html>
- [3] Beaufays, F.: The neural networks behind Google Voice transcription. August 11, 2015.
URL <http://googleresearch.blogspot.cz/2015/08/the-neural-networks-behind-google-voice.html>
- [4] Chansanchai, A.: Microsoft Research shows off advances in artificial intelligence with Project Adam. Júl 14, 2014.
URL <http://blogs.microsoft.com/next/2014/07/14/microsoft-research-shows-advances-artificial-intelligence-project-adam/>
- [5] Intel Xeon E5-4650L. 2015.
URL <http://www.techpowerup.com/cpudb/1686/xeon-e5-4650l.html>
- [6] Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. 18.01.2015.
URL <http://arxiv.org/pdf/1506.05751v1.pdf>
- [7] Fraser, N.: Neural Network Follies. September 1998.
URL <https://neil.fraser.name/writing/tank/>
- [8] Jia, Y.; Shelhamer, E.; Donahue, J.; aj.: Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, ACM, 2014, s. 675–678.
- [9] Kangas, L.: CATCH. Október 2000.
URL <https://www.ncjrs.gov/App/Publications/abstract.aspx?ID=190983>
- [10] Krewell, K.: CPU vs. GPU. December 16, 2009.
URL <http://blogs.nvidia.com/blog/2009/12/16/whats-the-difference-between-a-cpu-and-a-gpu/>

- [11] Metz, C.: Facebook's New AI Can Paint, But Google's Knows How to Party. 06.19.15.
URL <http://www.wired.com/2015/06/facebook-googles-fake-brains-spawn-new-visual-reality/>
- [12] Peter Auer, W. M., Harald Burgsteiner: A learning rule for very simple universal approximators consisting of a single layer of perceptrons. 18.1.2008.
URL <http://www.igi.tugraz.at/harry/psfiles/biopdelta-07.pdf>
- [13] Podyablonskiy, A.: Umělé neuronové sítě a jejich aplikace. Apríl 2013.
URL https://is.bivs.cz/th/15623/bivs_b/130426_BP_Podyablonskiy.pdf
- [14] Rosenblatt's Contributions. 2008.
URL <http://csis.pace.edu/~ctappert/srd2011/rosenblatt-contributions.htm>
- [15] Stergiou, C.; Siganos, D.: Neurónové siete.
URL http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Introduction%20to%20neural%20networks
- [16] Szegedy, C.; Liu, W.; Jia, Y.; aj.: Going Deeper with Convolutions.
URL <http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43022.pdf>
- [17] Templeton, G.: Neural network learns to paint. September 23, 2013.
URL <http://www.extremetech.com/computing/167179-facebook-is-working-on-deep-learning-neural-networks-to-learn-even-more-about-your-personal-life>
- [18] Venayagamoorthy, G. K.: TEACHING NEURAL NETWORKS CONCEPTS AND THEIR LEARNING TECHNIQUES.
URL https://www.asee.org/documents/sections/midwest/2004/Teaching_Neural_Networks_Concepts.pdf
- [19] Woodford, C.: Neural networks. 19.11.2014.
URL <http://www.explainthatstuff.com/introduction-to-neural-networks.html>

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

3D	trojrozmerný – three-dimensional
AI	umelá inteligencia – Artificial Intelligence
ANN	umelá neurónová sieť – Artificial Neural Network
BVLC	Berkeley Vision and Learning Center
CATCH	Computer Aided Tracking and Characterization of Homicides
CPU	Central Processing Unit
GMM	Gaussian Mixture Model
GPU	Graphics Processing Unit
LRN	Local Response Normalization
LSTM RNN	rekurentná neurónová sieť s dlhodobou a krátkodobou pamäťou – Long Short-term Memory Recurrent Neural Network
NYU	New York University
OS	operačný systém – Operation System
ReLU	Rectified Linear Unit
RNN	rekurentná neurónová sieť – Recurrent Neural Network

ZOZNAM PRÍLOH

A Obsah priloženého CD

38

A OBSAH PRILOŽENÉHO CD

CD obsahuje spustiteľný JAR súbor `xhladi19`. Ďalej obsahuje zložku `data`, ktorá obsahuje textové súboru s parametrami neurónových sietí a vyexportovaný archív `xhladi19` vo formáte `.zip` so zdrojovými kódmi. Taktiež obsahuje elektronickú formu bakalárskej práce.